

What is claimed is:

1 1. A system comprising:
2 a Unix operating system;
3 a plurality of execution entities;
4 an event control module adapted to create an event having a state, one or
5 more of the execution entities adapted to wait on the event; and
6 a controller adapted execution entities to awaken the one or more
7 execution entities by signaling the execution entities if the event state changes to a
8 predetermined state.

1 2. The system of claim 1, wherein the event control module is adapted to
2 define an event object representing the event, the event object associated with a queue
3 having one or more entries corresponding to one or more execution entities waiting on
4 the event represented by the event object.

1 3. The system of claim 2, wherein the event control module is adapted to
2 further create one or more second objects, wherein each entry comprises a link to a
3 corresponding second object, each execution entity to sleep on an associated second
4 object to wait on the event.

1 4. The system of claim 3, wherein each second object is defined by a
2 condition variable.

1 5. The system of claim 4, wherein the controller signals each thread by
2 signaling the condition variable.

1 6. The system of claim 3, wherein each second object is defined by a
2 condition variable and a mutex.

1 7. The system of claim 2, wherein each event object contains an indication of
2 the state of the event.

1 8. The system of claim 7, wherein the indication has a first state to indicate
2 that the event has been signaled and a second state to indicate that the event has not been
3 signaled, the predetermined state comprising the first state.

1 9. The system of claim 8, wherein each event object has a type indication to
2 indicate whether the event object state indication is to be automatically reset to the
3 second state from the first state once the event has been signaled or to be manually reset
4 to the second state from the first state by an explicit action.

1 10. The system of claim 1, wherein the event control module is adapted to
2 further define at least another event, one of the execution entities to wait on the plural
3 events.

1 11. The system of claim 10, wherein each of the events is represented by a
2 corresponding event object, each event object having a first state to indicate that the event
3 has been signaled and a second state to indicate that the event has not been signaled.

1 12. The system of claim 11, further comprising queues associated with
2 corresponding event objects, each queue containing an entry corresponding to the one
3 execution entity.

1 13. The system of claim 12, wherein the event control module is adapted to
2 define a barrier object, the one execution entity to sleep on the barrier object to wait on
3 the plural events, the entries of the queues each containing a link to the barrier object.

1 14. The system of claim 13, wherein the barrier object is defined at least by a
2 condition variable.

1 15. The system of claim 13, wherein the barrier object is defined at least by a
2 condition variable and a mutex.

1 16. The system of claim 1, wherein the event control module comprises a
2 library.

1 17. The system of claim 1, wherein the execution entities comprise threads.

1 18. The system of claim 17, further comprising plural processes, each process
2 associated with one or more threads,
3 the event control module to create a local event to synchronize threads
4 within a process and to create a global event to synchronize threads of different
5 processes.

1 19. The system of claim 18, wherein the global event comprises a named
2 event.

1 20. The system of claim 1, further comprising a plurality of nodes, each node
2 comprising one or more of the plurality of execution entities.

1 21. An article comprising at least one storage medium containing instructions
2 for providing event-based synchronization in a system in which execution entities are
3 running, the instructions when executed causing the system to:
4 generate event objects representing events used for synchronizing
5 execution entities in the system, each event object having a state to indicate if the
6 corresponding event has been signaled;
7 provide one or more entries associated with the event object, each entry
8 associated with a corresponding execution entity; and
9 in response to the state of one of the event objects indicating the
10 corresponding event has been signaled, use the one or more entries to signal one or more
11 corresponding execution entities.

1 22. The article of claim 21, wherein the instructions when executed cause the
2 system to further create barrier objects, each execution entity waiting on a corresponding
3 barrier object to wait on an event.

1 23. The article of claim 22, wherein the instructions when executed cause the
2 system to create barrier objects by defining each barrier object based on a condition
3 variable according to a Unix operating system.

1 24. The article of claim 22, wherein the instructions when executed cause the
2 system to create barrier objects by defining each barrier object based on a condition
3 variable and mutex according to a Unix operating system.

1 25. The article of claim 21, wherein the instructions when executed cause the
2 system to define a queue associated with each event object, the queue containing the one
3 or more entries, the one or more entries pointing to one or more barrier objects.

1 26. The article of claim 25, wherein the instructions when executed cause the
2 system to provide a routine associated with each event object, the routine to traverse each
3 queue and to signal one or more barrier objects pointed to by one or more entries in the
4 queue.

1 27. The article of claim 21, wherein the instructions when executed cause the
2 system to provide plural events containing respective entries, each of the entries
3 corresponding to one execution entity to enable the one execution entity to wait on plural
4 events.

1 28. A method of providing event-based synchronization in a system having
2 plural execution entities, comprising:
3 providing one or more synchronization primitives;
4 defining a first object based on the one or more synchronization
5 primitives;

6 defining an event object representing an event, the event object having a
7 state to indicate the event being signaled; and
8 one of the execution entities sleeping on the first object to wait on the
9 event.

1 29. The method of claim 28, further comprising signaling the first object in
2 response to the event object state indicating the event being signaled.

1 30. The method of claim 29, wherein signaling the first object comprises a
2 routine associated with the event object signaling the first object.

1 31. The method of claim 30, wherein providing the one or more
2 synchronization primitives comprises providing one or more synchronization primitives
3 defined in a Unix operating system.

1 32. The method of claim 31, wherein the one or more synchronization
2 primitives comprises a condition variable, wherein signaling the first object comprises
3 signaling the condition variable.

1 33. The method of claim 28, wherein providing the one or more
2 synchronization primitives comprises providing one or more synchronization primitives
3 defined in a Unix operating system.

1 34. The method of claim 28, further comprising defining at least another event
2 object representing another event, the one execution entity to add entries to the event
3 objects to enable the one execution entity to wait on plural events.

1 35. The method of claim 34, wherein the one execution entity adding the
2 entries to the event objects comprises adding entries to queues associated with the event
3 objects.

1 36. The method of claim 35, wherein adding the entries to the queues
2 comprises adding a pointer to the first object.

1 37. A system comprising:
2 a Unix operating system;
3 a plurality of execution entities;
4 a storage module containing an event library; and
5 a processor adapted to execute the event library to provide an event-based
6 synchronization mechanism comprising one or more events on which the plural execution
7 entities are able to sleep.

1 38. The system of claim 37, further comprising plural processes, each process
2 associated with one or more of the execution entities, wherein the synchronization
3 mechanism comprises a local event synchronization mechanism to synchronize execution
4 entities associated with one process, and a global event synchronization mechanism to
5 synchronize execution entities associated with plural processes.